DTIC FILE COPY

ARI Research Note 87-04

AD-A179 992

ADDRESSING THE TRAVELLING SALESMAN PROBLEM THROUGH EVOLUTIONARY ADAPTATION

David B. Fogel
Titan Systems

for

Contracting Officer's Representative
George H. Lawrence

BASIC RESEARCH LABORATORY
Milton R. Katz, Director

DTIC
ELECTE
APR 2 9 1987
S D
E

U. S. Army
Research Institute for the Behavioral and Social Sciences

January 1987

Approved for public release; distribution unlimited.

87 4 28 141

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

WM. DARRYL HENDERSON
COL, IN
Commanding

EDGAR M. JOHNSON
Technical Director

Research accomplished under contract
for the Department of the Army

Titan Systems

Technical review by

Steve Kronheim

This report, as submitted by the contractor, has been cleared for release to Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or other reference services such as the National Technical Information Service (NTIS). The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARI Research Note 87-04	2. GOVT ACCESSION NO. AD-A179	3. RECIPIENT'S CATALOG NUMBER 992
4. TITLE (and Subtitle) Addressing the Travelling Salesman Problem Through Evolutionary Adaptation		5. TYPE OF REPORT & PERIOD COVERED Interim Report April - July 1986
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David B. Fogel		8. CONTRACT OR GRANT NUMBER(s) PO-9-X56-1102C-1
9. PERFORMING ORGANIZATION NAME AND ADDRESS Titan Systems P.O. Box 12139 La Jolla, CA 92037		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (Not Applicable)
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333-5600		12. REPORT DATE January 1987
		13. NUMBER OF PAGES 36
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) - -		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE - -

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

- -

18. SUPPLEMENTARY NOTES

George H. Lawrence, contracting officer's representative

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Algorithm — Evolutionary Programming
Problem Solving — Iterative Mutation
Logic

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Optimizing the "travelling salesman" problem continues to receive attention for three reasons: its solution is computationally difficult, although the algorithm itself is easily expressed; it is broadly applicable to a variety of engineering problems; and it has become somewhat of a comparison "benchmark" problem.

The task contained in the problem is to arrange a tour of n cities in such a way that each city is visited only once, and the length of the tour (or some

(OVER)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ARI Research Note 87-04

20. Abstract (continued)

other cost function) is minimized. For an exact solution, the only known algorithms require the number of steps to grow at least exponentially with the number of elements in the problem. Brute force methods of finding the shortest path require the compilation of a list of (n-1)!/2 alternative tours, a number which grows faster than any finite power of n, so the task quickly becomes unmanageable.

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification

By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

QUALITY INSPECTED 1

ADDRESSING
THE TRAVELING SALESMAN PROBLEM
THROUGH EVOLUTIONARY ADAPTATION

By David Fogel

INTRODUCTION

The optimization of the traveling salesman problem continues to receive attention for three reasons: (1) its solution is computationally difficult although the algorithm itself is easily expressed; (2) it is broadly applicable to a variety of engineering problems; and (3) it has become somewhat of a comparison "benchmark" problem. The task is to arrange a tour of n cities such that each city is visited only once and the length of the tour (or some other cost function) is minimized. For an exact solution the only known algorithms require the number of steps to grow at least exponentially with the number of elements in the problem. Brute force methods of finding of the shortest path by which a traveling salesman can complete a tour of n cities requires compiling a list of (n-1)!/2 alternative tours, a number that grows faster than any finite power of n. The task quickly becomes unmanageable.

BACKGROUND

Two recent papers (Goldberg, Lingle, Jr., 1985; Grefenstette et al., 1985) addressed the traveling salesman problem through use of the genetic algorithm as proposed by Holland (1975). This algorithm is an offshoot of the evolutionary programming concept offered by Fogel (1962, 1964, Fogel et al., 1966).

In Fogel's evolutionary procedures the process of iterative mutation and selection is simulated to evolve a logic most suitable for resolving the problem at hand. Intelligent behavior is viewed as requiring prediction of an environment coupled with the use of such predictions for the sake of controlling that environment (to the greatest extent possible). The behavior of each artificial organism is constructed as a finite state machine, a general mathematical function that does not constrain the represented transduction to be linear, passive, or without hysteresis.

The evolutionary process is simulated in the following manner: an original "machine" (an arbitrary logic or a "hint") is measured in its ability to predict each next event in its "experience" with respect to whatever payoff function has been prescribed. Progeny are now created through random mutation of this "parent" machine. They are scored in a similar manner to the parent in predictive ability. If the parent is better than its offspring, the parent is used to generate other offspring. If, however, an offspring is better than its parent, that offspring becomes the new parent.

This assures non-regressive evolution. An actual prediction is made when the predictive fit score demonstrates that a sufficient level of credibility has been achieved. The surviving machine generates a prediction, indicates the logic of this prediction and becomes the progenitor for the next sequence of progeny, this in preparation for the next prediction. Thus, randomness is selectively incorporated into the surviving logic. The sequence of predictor machines demonstrates phyletic learning, an inductive generation of sequences of hypotheses concerning the relevant regularities found within the experienced environment, in the context of the given payoff function.

Holland's approach differs from that of Fogel's. Rather than describe each organism only in terms of its behavior, Holland emphasizes the coding structures which generate such organisms. Holland's genetic algorithms search a parameter space where "any point in the parameter space can be represented as an n bit vector." "There are two primary operations applied to the population by a genetic algorithm. Reproduction changes the contents of the population by adding copies of genotypes with above-average figures of merit." "Crossover is the primary means of generating plausible new genotypes for addition to the population" (Ackley, 1985).

Holland defines crossover as taking two coding structures, $A_1 = a_{11}a_{12}...a_{1n}$ and $A_2 = a_{21}a_{22}...a_{2n}$, and at a random point "x" between 1 and n, exchanging the set of attributes to the right of this position yielding offspring of the form: $A^* = a_{11}a_{12}...a_{1x}a_{2(x+1)}...a_{2n}$. "This 'offspring' is added to the population, displacing some other genotype

according to various criteria where it has the opportunity to flourish or perish depending on its fitness. Mutation provides a chance for any allele to be changed to another randomly chosen value. If the mutation rate is too low, possibly critical alleles missing from the initial population will have only a small chance of getting...into the population. However, if the probability of a mutation is not low enough, information...will be steadily lost to random noise" (Ackley, 1985).

Holland likens the actual code being mutated to that of the genetic code that defines a natural organism. While Fogel et al. (1966) only used small degrees of "background" mutation, Holland incorporates the operations of gene "crossover" and "inversion" among other actual biologic genetic recombinations. Although Holland's work has gone largely unnoticed for some time, today renewed attention is being given to genetic algorithms.

Goldberg and Lingle (1985) offered several observations of the genetic algorithm (GA) as it relates to the traveling salesman problem:

"1) Simple genetic algorithms work well in problems which can be coded so the underlying building blocks (highly fit, short defining length schemata) lead to improved performance.

"2) There are problems (more properly codings for problems) that are GA-hard -- difficult for the normal reproduction + crossover + mutation processes of the simple genetic algorithm.

"3) Inversion is the conventional answer when genetic algorithmists

are asked how they intend to find good string ordering, but inversion has never done much in empirical studies to date.

"4) Despite numerous rumored attempts, the traveling salesman problem has not succumbed to genetic algorithm-like solution."

They suggested a new type of crossover operator, the "partially-mapped crossover (PMX)," which they believe will lead to a more efficient solution of the traveling salesman problem.

PMX would proceed as follows: consider two possible codings of a tour of eight cities, A_1 and A_2, a return to the initial city being implicit:

A_1: 3 5 1 2 7 6 8 4

A_2: 1 8 5 4 3 6 2 7

Two positions are determined randomly along the A_1 coding. The actual cities located between these positions along A_1 are exchanged with the cities located between the same positions along A_2. For example, if the positions three and five are chosen, the sub-coding along A_1 is 1-2-7, and the sub-coding along A_2 is 5-4-3. Each of these cities is then exchanged, leading to the new tours, A^*_1 and A^*_2:

A^*_1: 7 1 5 4 3 6 8 2

A^*_2: 5 8 1 2 7 6 4 3.

Goldberg and Lingle (1985) reported two experiments on ten cities where the PMX operator enabled the search to efficiently discover either the absolute or near optimum solution.

Grefenstette et al. (1985) addressed the traveling salesman problem using Holland's "simple crossover." This required the formation of a special coding structure. Clearly, using this operator on two valid tours could result in an "offspring" that was not a valid tour. As Dewdney (1985) has commented, the authors' method for devising the appropriate coding was ingenious.

> "The representation for a five-city tour such as *a, c, e, d, b* turns out to be 12321. To obtain such a numerical string reference is made tc some standard order for the cities, say, *a, b, c, d, e*. Given a tour such as *a, c, e, d, b*, systematically remove cities from the standard list in the order of the given tour: remove *a*, then *c*, *e* and so on. As each city is removed from the special list, note its position just before removal, *a* is first, *c* is second, *e* is third, *d* is second and, finally *b* is first. Hence the chromosome 12321 emerges. Interestingly, when two such chromosomes are crossed over, the result is always a tour."

Unfortunately the experiments with this representation were "not very encouraging" (Dewdney, 1985). Grefenstette et al. conducted larger experiments than those of Goldberg and Lingle, including 50, 100 and 200 cities. In the three reported experiments, after a large number of trials (approximately 14000, 20000 and 25000, respectively), the best tours were still far away from the expected optimal solutions.

At this point it is natural to ask "why?". After all, the traveling salesman problem only requires discovery of a logical pattern. This seems completely analogous to what occurs in nature. If the crossover of genes works in natural evolution, why shouldn't it work here?

The answer is, in fact, that suggested by Goldberg and Lingle's second observation: the traveling salesman problem is difficult to address using Holland's crossover mutation. This is because the crossover operation, as defined by Holland, does not mimic the biological crossover of genes. Natural crossover is a phenomenon where "old linkages between genes on homologous chromosomes are broken and new linkages are established. Genes that reside on the same chromosome and move together are said to be 'linked.' A linkage group is any group of genes physically linked on one chromosome...Changes in linkage groups are not truly mutations, however, since neither the amount nor the function of genetic material is altered" (Levy, 1982).

Holland's crossover treats the entire tour as a chromosome and each city in a tour somewhat as a gene. While this does not change the amount of coding, it greatly alters the function of the coding. Natural crossover allows for different combinations of alleles. Alleles, by definition, control the same characteristic and occupy the same place on similar chromosomes. A more appropriate biologic interpretation of a tour would be that it is itself a gene. Crossover inside a gene is a nonsequetor. The tour is not analogous to a chromosome and each city in a tour is not analogous to a gene. These relations are in fact anomolous.

The result of Holland's crossover is therefore a near random search throughout the entire space of possible tours. This is, of course, the essence of the difficulty. Dewdney (1985) has commented that by using Holland's crossover "there is so much juggling of genes and cracking of chromosomes that...(a parent)...is hard put to recognize its own grandchildren." As the number of cities grows larger, Holland's crossover effectively destroys the link between each parent and its offspring. The results can even be worse than a complete enumeration of all possible tours (Appendix). Adaptive plans must retain previous advances and incorporate them into future solutions.

AN ALTERNATIVE APPROACH

An alternative solution is the adaptive algorithm, so named because it does not include any of the genetic-mimicking operators that Holland has suggested, but instead emphasizes the behavioral appropriateness (fitness) of the evolved trial solutions. The algorithm, which is equivalent to Fogel's evolutionary programming restricted to single state machines, only slightly mutates the existing tour by removing just one city from a given list and replacing it in a different randomly chosen position. This mutation is only mildly more complicated than the simplest possible mutation, that is, swapping adjacent cities. It is clearly less complex than either the PMX operator or Holland's crossover; through multiple mutation, this single alteration can be made equivalent to either of these crossover operators. Holland (1975) has stated: "If successive populations are produced by mutation alone (without (genetic) reproduction), the result is

a random sequence of structures drawn from (all possible structures)." This is only partially correct. The adaptive algorithm does result in a random search, but only in that portion of the space relatively close to the parent which generates the offspring. This dramatically increases the effectiveness of the search through the state space of all possible constructions.

Not only must advances be retained but "dead-ends" must be circumvented. Because there is a finite number of offspring that can be generated through mutation evolutionary stagnation might well occur on a local optimum. To prevent this it is useful to randomly alter the adaptive topography (payoff function) that is being searched. This can be accomplished by a variety of means. One of these is to occasionally allow for the survival of offspring that are slightly worse than their parents. In effect, the scoring function is made "noisy."

What results is analogous to the searching of a maze; when a dead-end is reached some backtracking is allowed and the overall search is reinitiated. Unfortunately, the topography is much like an upside-down bed of nails, with some nails being longer (better) than others. From any given nail, it is possible to travel to n(n-1) other nails in a single mutation by randomly choosing a city and placing it in a different position. Unlike a maze, when the evolving phyletic line reaches a non-optimal nail from which no single mutation results in a better tour, it is impossible to determine the "direction" in which to backtrack. The complete prevention of evolutionary stagnation is impossible unless all inheritance is given up and the search made completely random.

EXPERIMENTAL FINDINGS

Experiments were performed to determine the effectiveness of the adaptive algorithm. Initially, 128 independent trials were performed on a 24 city traveling salesman problem where the cities were positioned on the periphery of a rectangle. Clearly, the minimum length tour is equal to the perimeter of the rectangle, in this case, 250. The amount of noise that was used is indicated in Table 1. The same degree of noise was used throughout all of the experiments described. Of the 128 trials performed,

---

Table 1:

Number of Evaluated Offspring	Offspring Score/Parent Score	Accept%
Less than 1,500	<= 1.05	15%
	<= 1.1	10%
	<= 1.2	5%
	> 1.2	1%
Between 1,500 and 5,000	<= 1.05	5%
	<= 1.1	2.5%
	<= 1.2	1%
	> 1.2	0.5%
Between 5,000 and 10,000	<= 1.05	2%
	<= 1.1	1%
	<= 1.2	0.5%
	> 1.2	0.2%
Between 10,000 and 20,000	<= 1.05	0.5%
	> 1.05	0%
Greater than 20,000	Any ratio	0%

Table 1: The amount of noise.

90.625% found the optimum solution in an average of 5297.48 iterations (Figure 1) where the maximum number of iterations was arbitrarily set at 14,000. Figure 2 indicates the results of the remaining 9.375% of the trials in which the evolving tours were, at least temporarily, trapped on a local optimum. Despite the seemingly non-complex arrangement of cities, the numerous local optima inherent to this city-structure make this particular traveling salesman problem somewhat recalcitrant.

To further investigate the efficiency of the algorithm, 20 experiments were conducted requiring a tour of 50 cities where the cities were redistributed for each experiment. In each, no optimum tours were discovered in 20,000 iterations, but it was clear that the evolutionary process was "solving the problem." Figure 3 indicates the results of a typical experiment. Figure 4 indicates the mean and estimated two-sigma limits of the evolutionary process as it discovered more and more suitable tours as offspring were evaluated. Note that "backtracking" played an integral part of the search.

Experiments were then performed requiring a tour of 100 cities under similar conditions. Again, while none of the eight experiments found a perfect tour in 20,000 iterations, the evolutionary process performed well. Figure 5 indicates the results of a typical experiment while Figure 6 indicates the mean and two-sigma limit of the reduction in tour length as offspring were evaluated.

Further experiments required a tour of 90 cities. Here, 18 trials

Example of Successful Discovery of the Optimum Solution in Series #1

Experiment #81

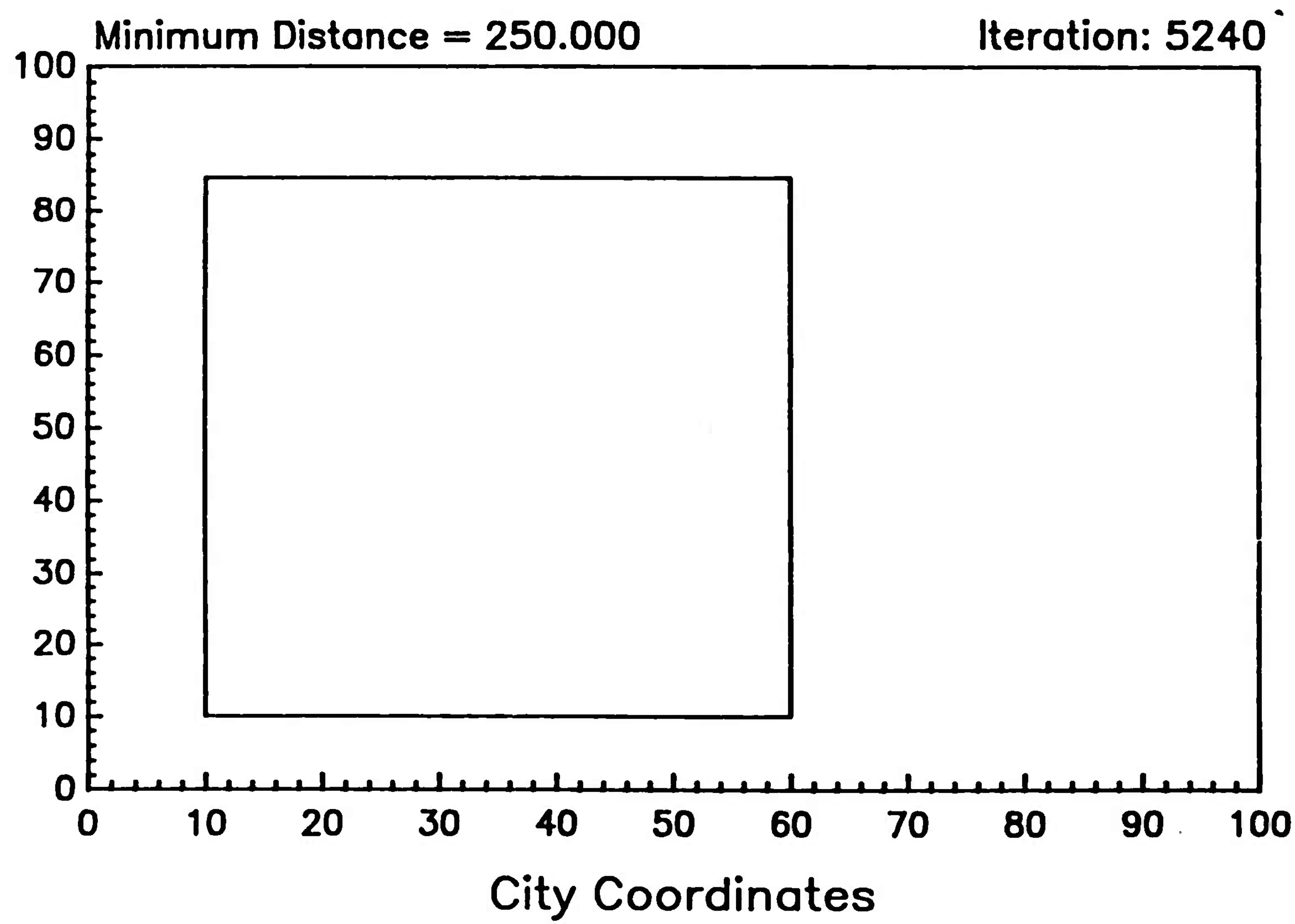


Figure 1

Typical Example of Stagnation in Series #1

Experiment #76

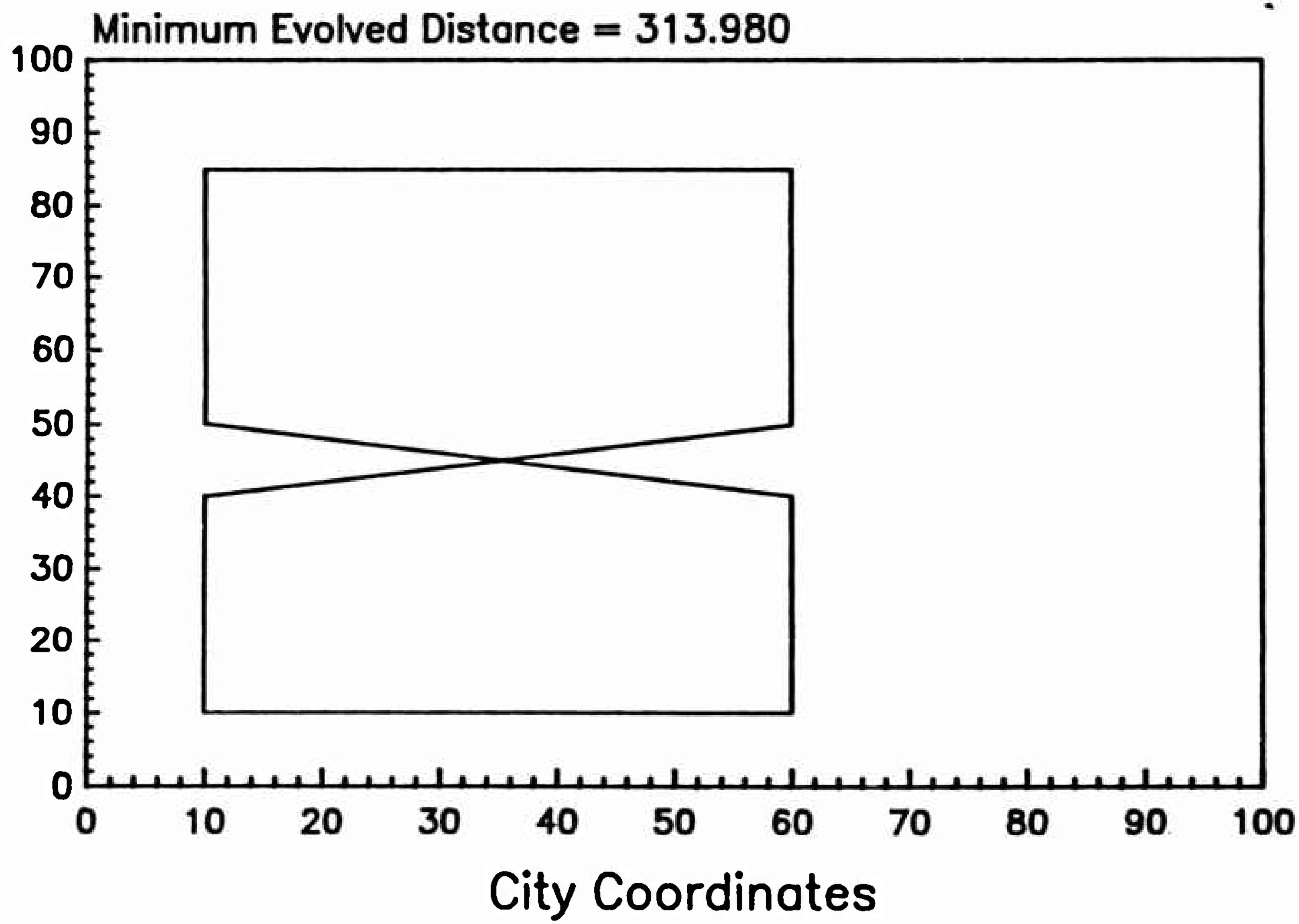
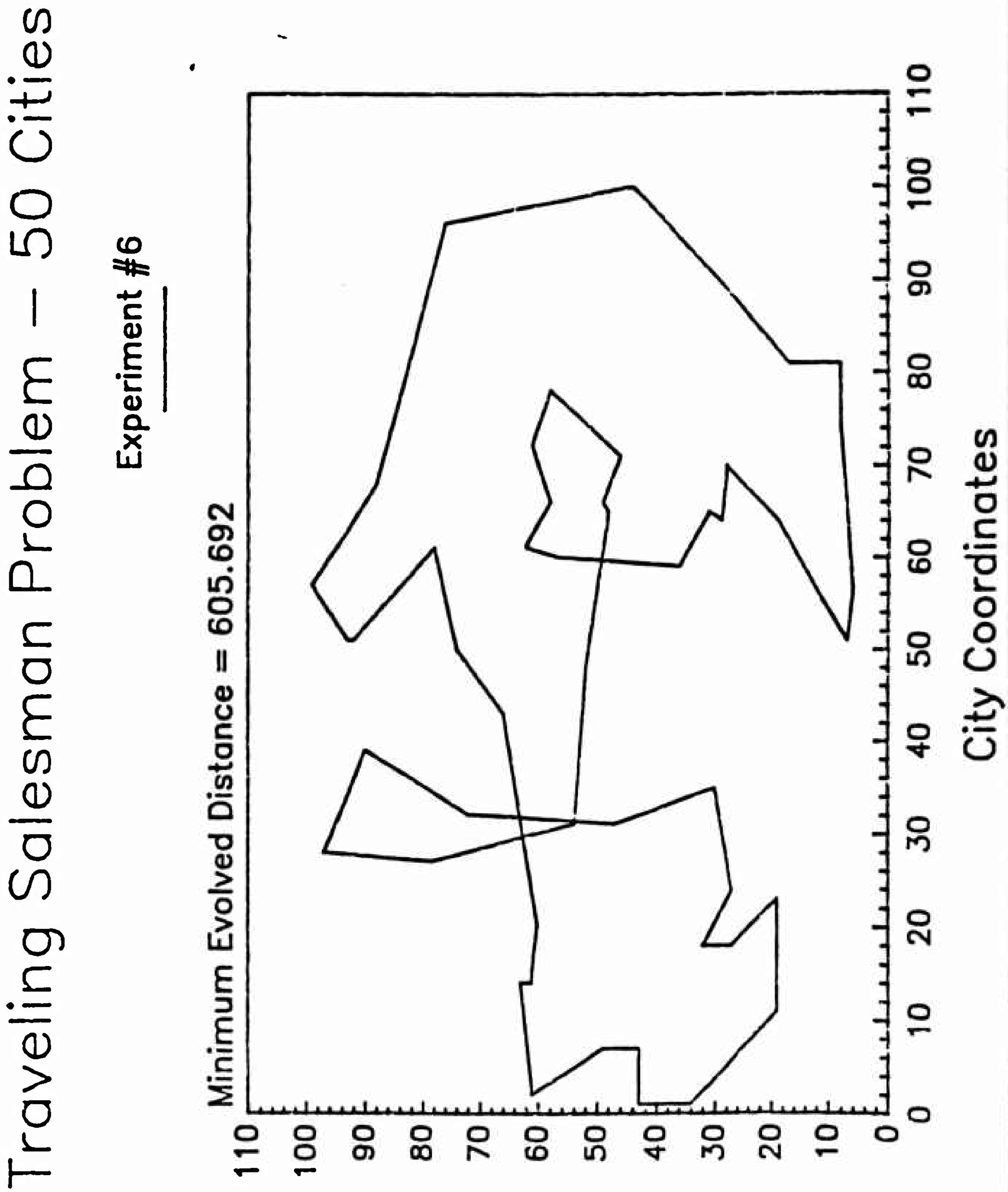
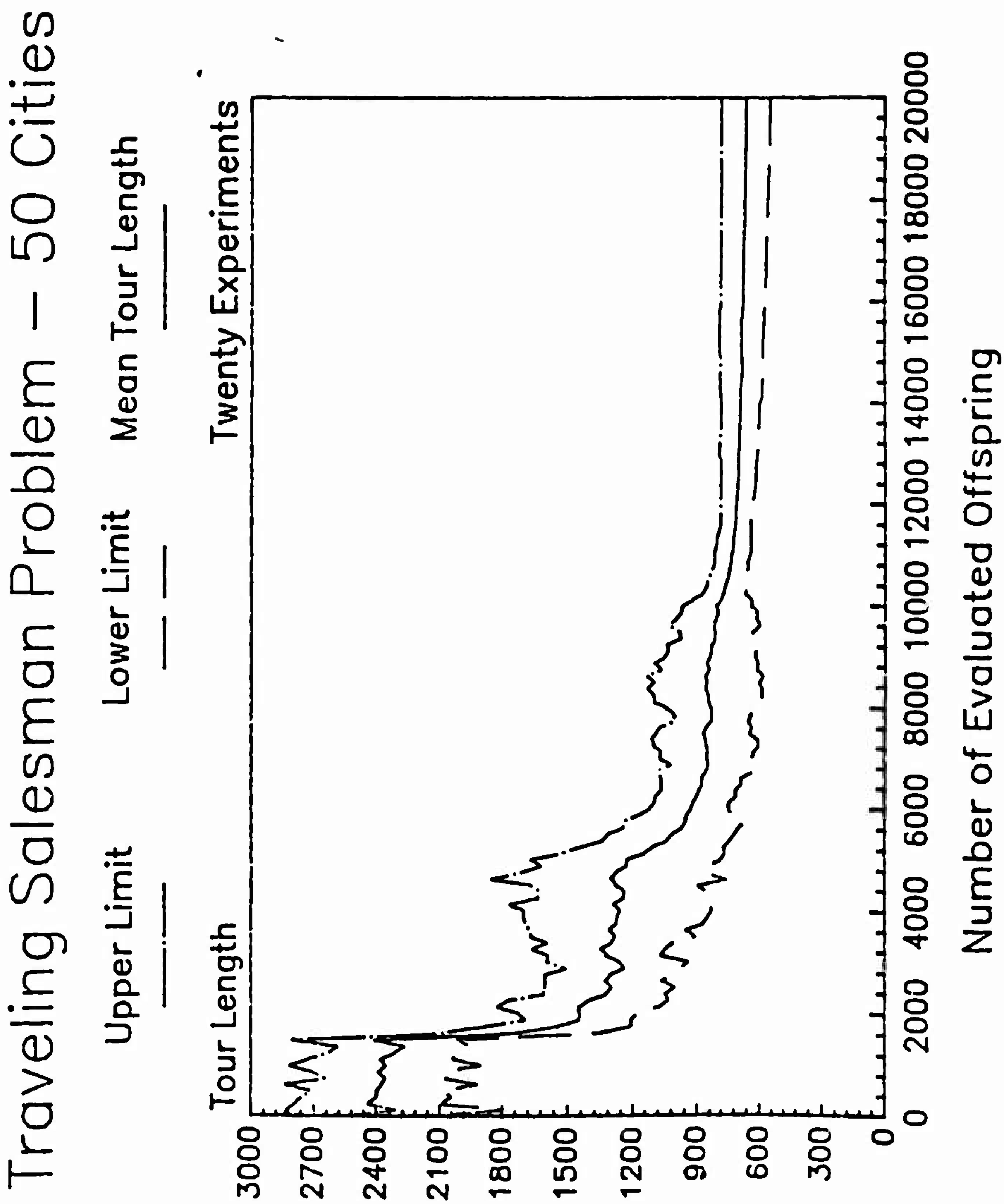


Figure 2



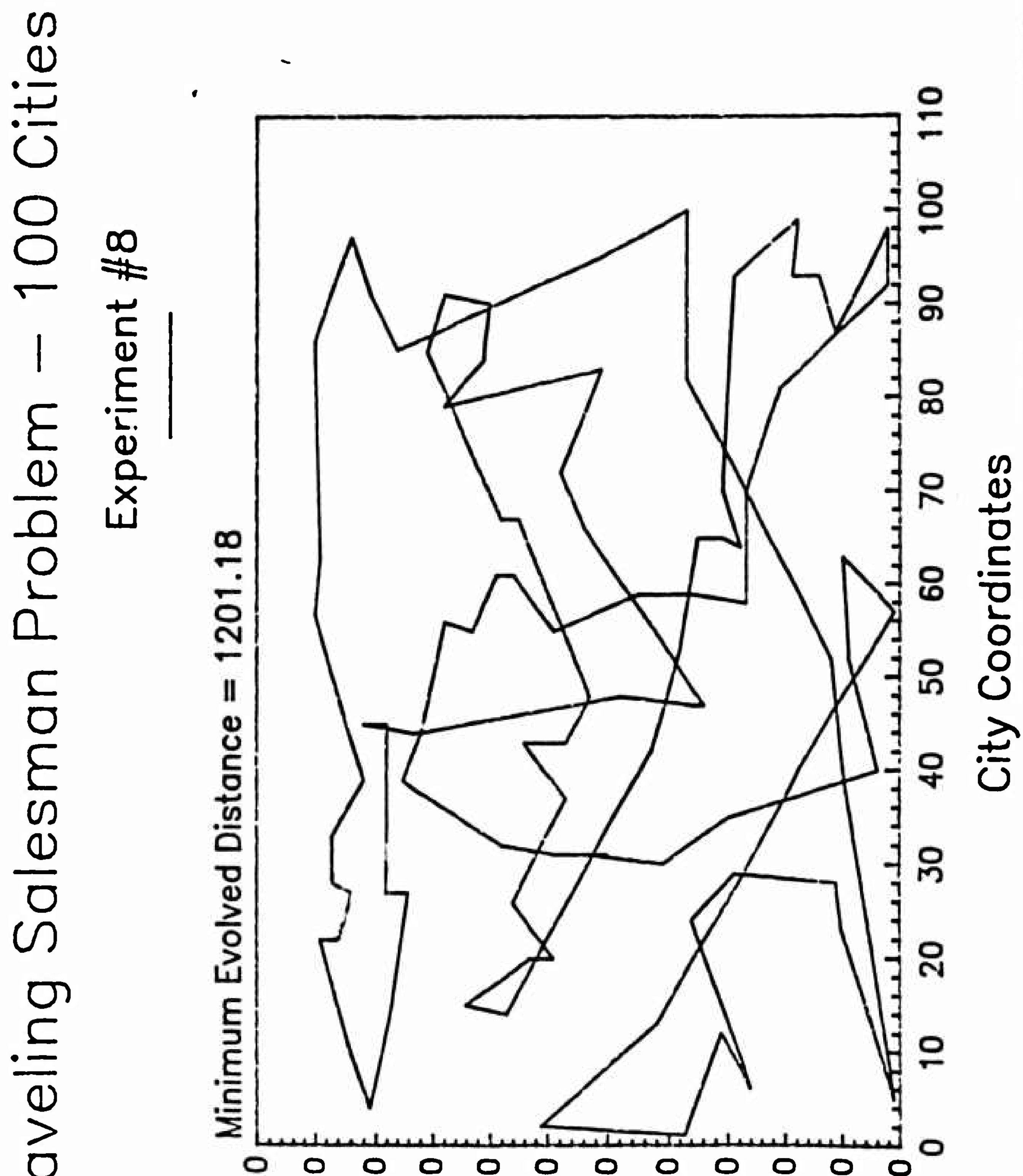
Traveling Salesman Problem — 50 Cities
Experiment #6
Minimum Evolved Distance = 605.692
City Coordinates
0
10
20
30
40
50
60
70
80
90
100
110

Figure 3



Traveling Salesman Problem – 50 Cities
Upper Limit
Lower Limit
Mean Tour Length
Tour Length
Twenty Experiments
3000
2700
2400
2100
1800
1500
1200
900
600
300
0
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
Number of Evaluated Offspring

Figure 4



Traveling Salesman Problem — 100 Cities
Experiment #8
Minimum Evolved Distance = 1201.18
110
100
90
80
70
60
50
40
30
20
10
0
City Coordinates

Figure 5

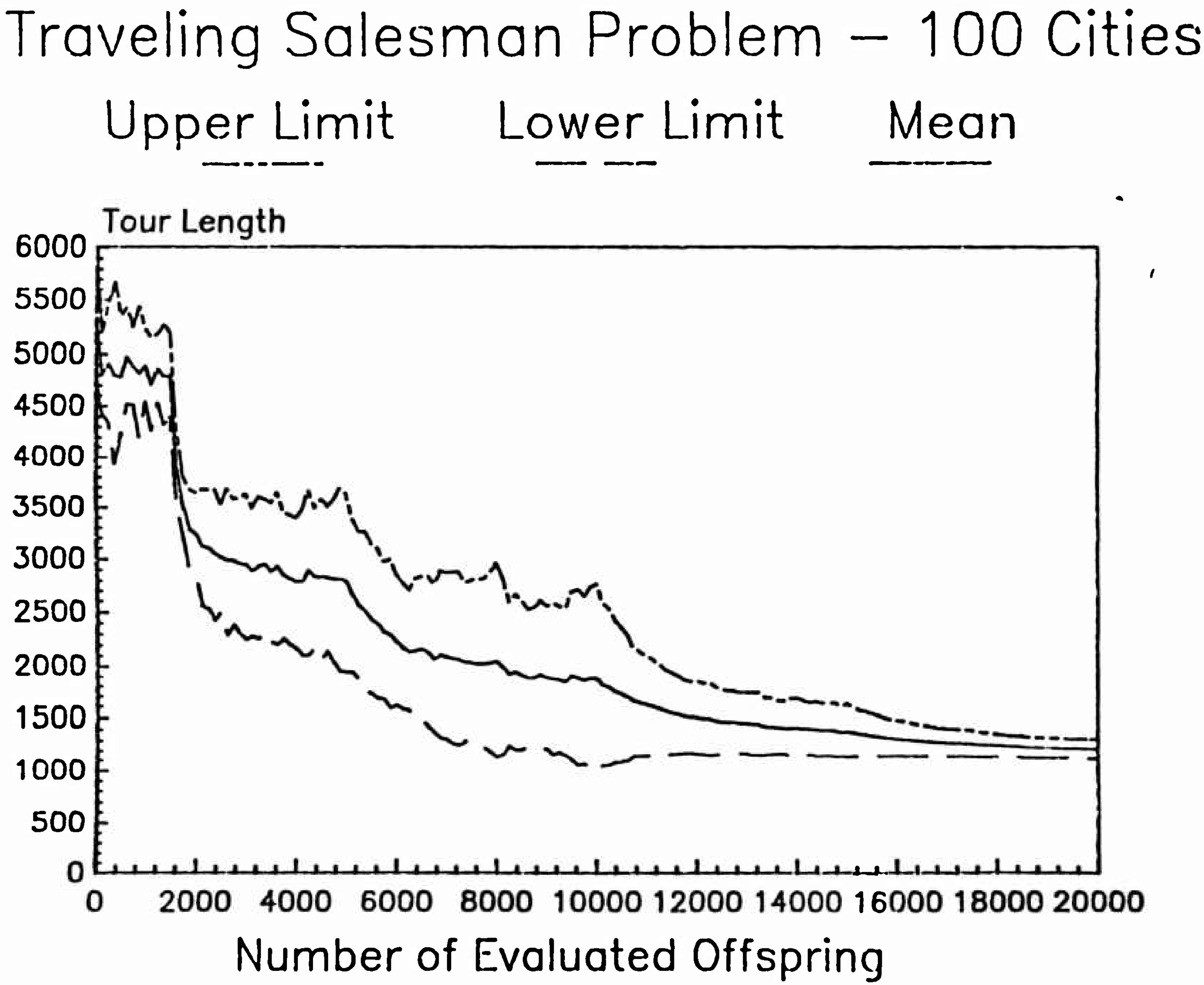
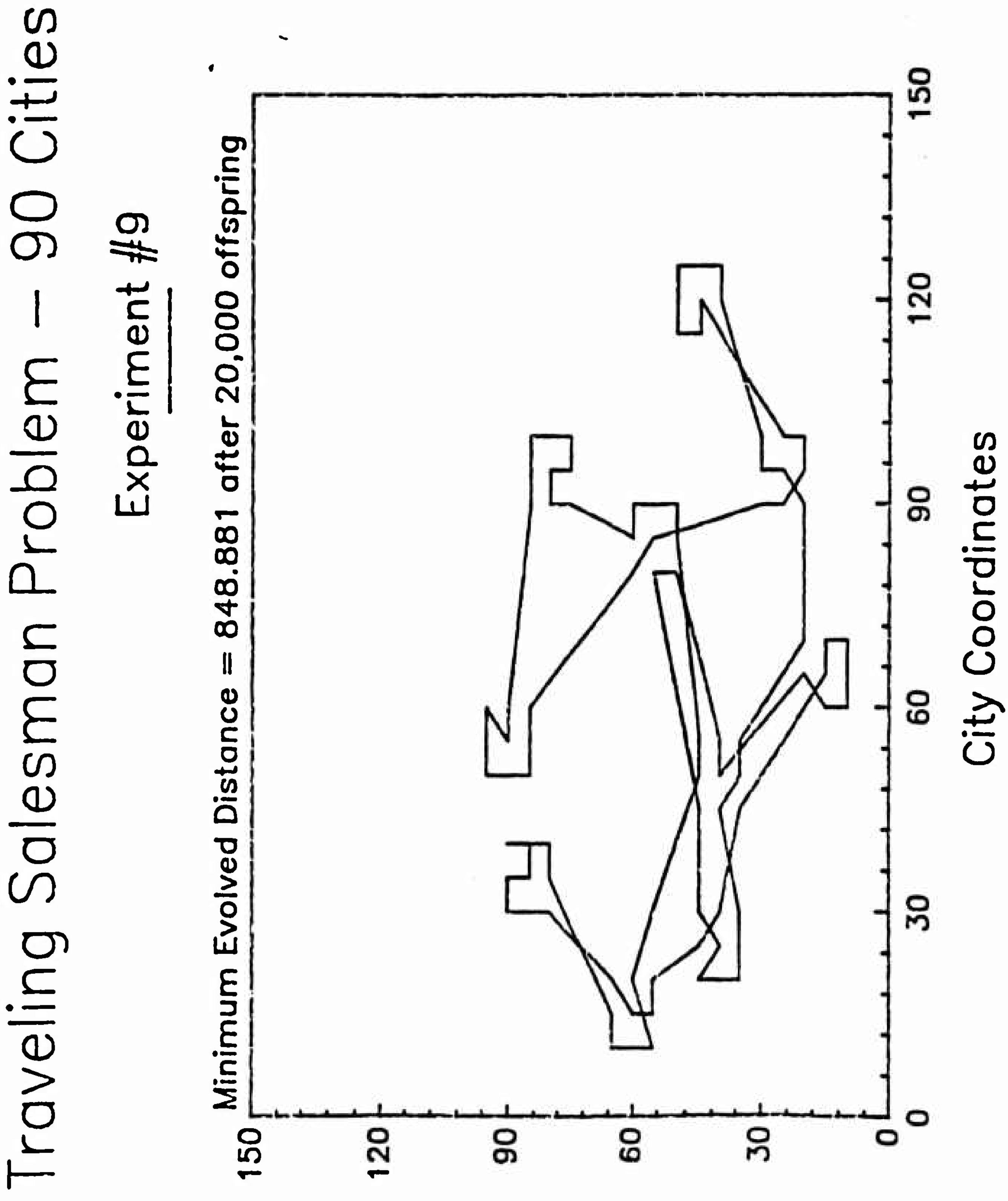

Traveling Salesman Problem — 100 Cities
Upper Limit
Lower Limit
Mean
Tour Length
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
Number of Evaluated Offspring

Figure 6

were performed on ten groups of nine cities that were randomly placed on the coordinate grid. The process was allowed to evolve 32,000 offspring. While the optimum solution remained undiscovered, it is of interest to note that the problem was evidently addressed at two distinct levels. The evolutionary process initially solved the problem at a gross level, discovering the minimum tour between the groups of cities (Figure 7 and Figure 8). Insufficient time was allowed to sort out the problem at a finer level of detail. Figure 9 indicates the mean and estimated two-sigma limits to the reduction of tour length up to the 20,000th iteration.

An extremely large traveling salesman problem was also analyzed. Here, 256 cities were randomly distributed. Based on the previous results it was not expected that the adaptive algorithm would discover the optimum solution in 20,000 iterations; however, after only 10,000 iterations it had reduced the initial tour length by roughly 50 percent. Figure 10 indicates the "surviving" tour after evaluating 10,000 offspring while Figure 11 indicates the success of the evolutionary process in discovering better and better tours. The available computation time limited the analysis, however the results were certainly encouraging.

The evolutionary program was also extended to allow for cities distributed in three dimensions. Here, 50 cities were randomly distributed. As expected, the addition of the third dimension had little effect on the adaptive algorithm. The initial tour length was reduced by 50 percent in fewer than 6,000 iterations, see Figure 12. Figure13 indicates two views of the surviving tour after evaluating 20,000 offspring.



Traveling Salesman Problem – 90 Cities
Experiment #9
Minimum Evolved Distance = 848.881 after 20,000 offspring
150
120
90
60
30
0
0
30
60
90
120
150
City Coordinates

Figure 7

Traveling Salesman Problem – 90 Cities

Experiment #9

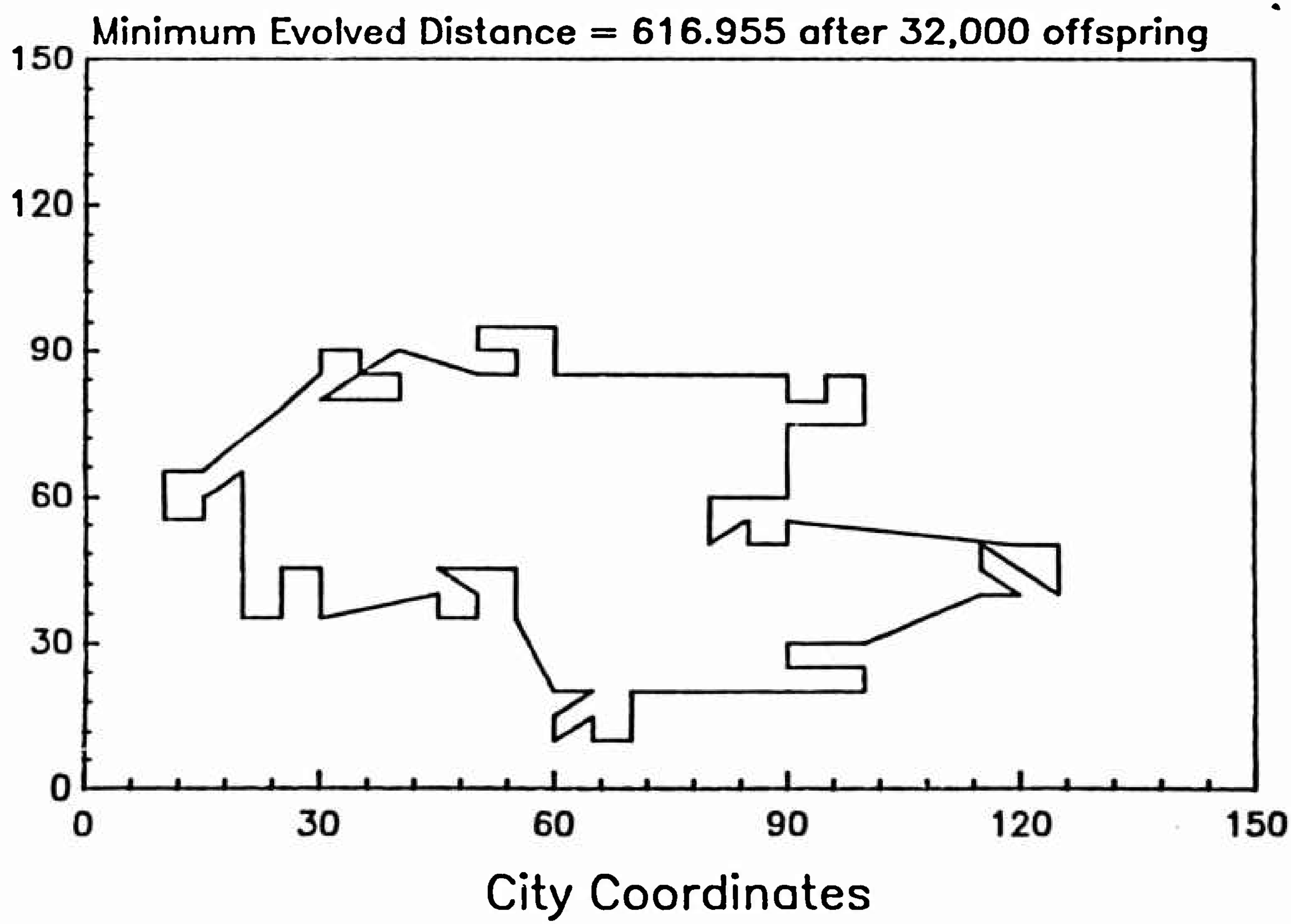


Figure 8

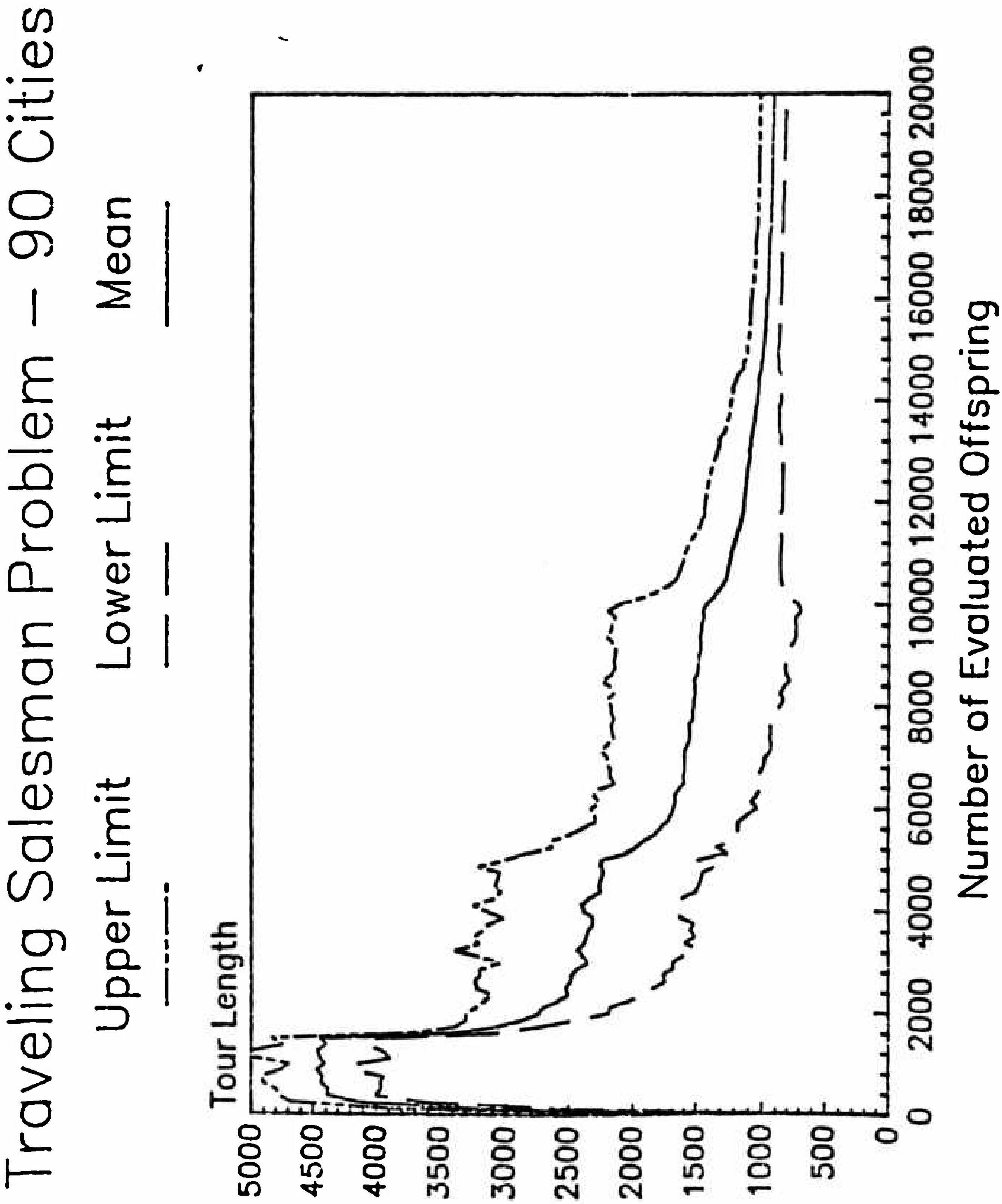

Traveling Salesman Problem – 90 Cities
Upper Limit
Lower Limit
Mean
Tour Length
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
Number of Evaluated Offspring

Figure 9

Traveling Salesman Problem – 256 cities

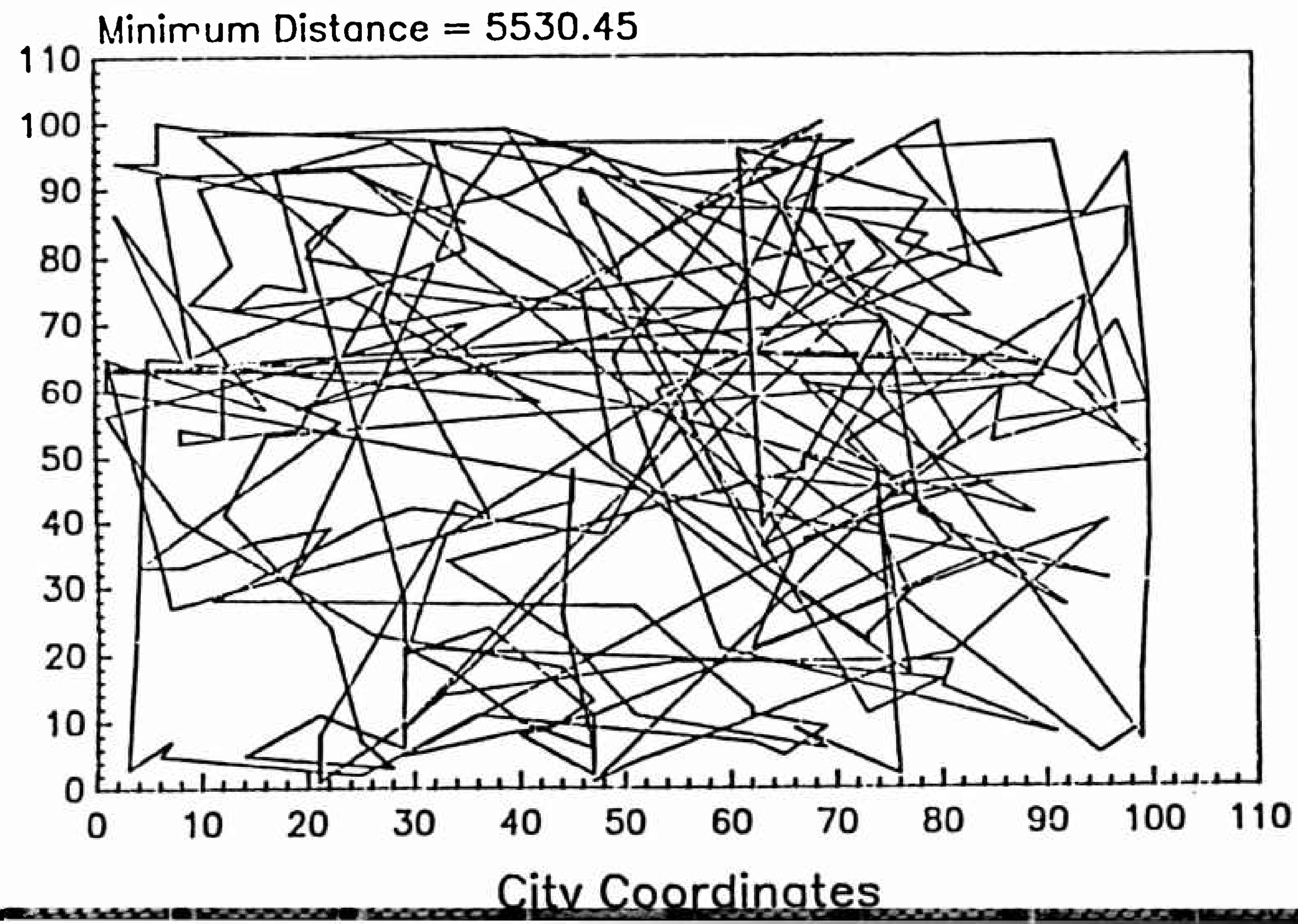


Figure 10

Traveling Salesman Problem — 256 cities

Evolutionary Improvement

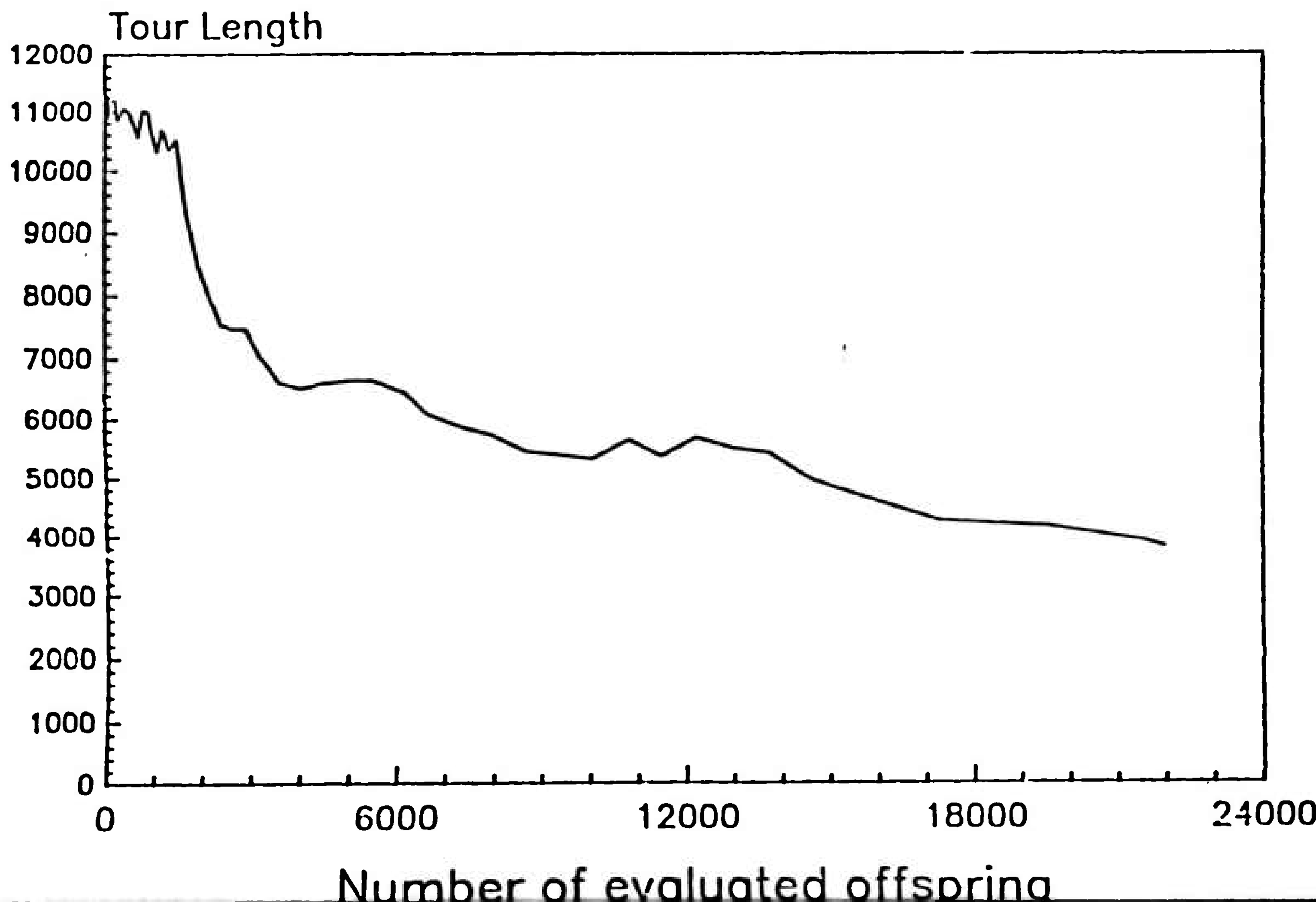
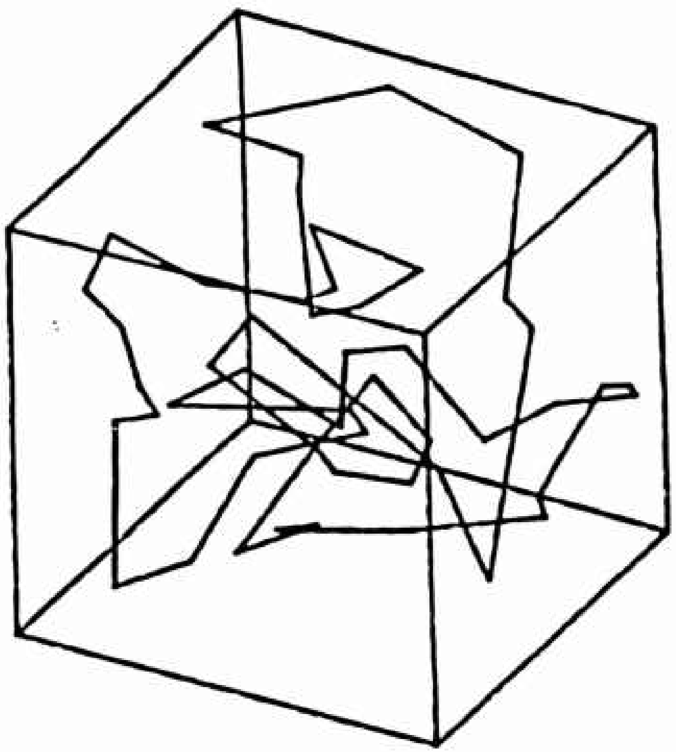


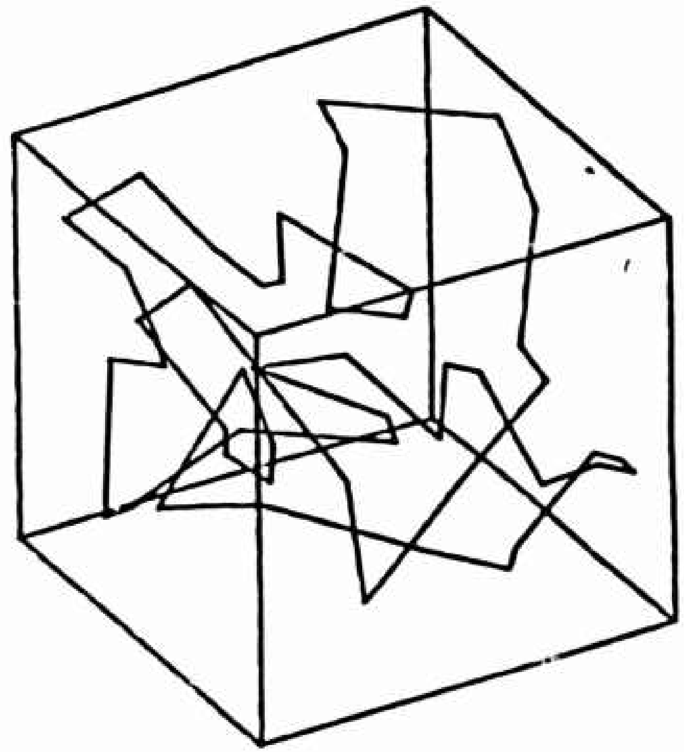
Figure 11

Traveling Salesman Problem

50 Cities Distributed in Three Dimensions



Rotation: 30 degrees



Rotation: 60 degrees

Final Evolved Tour After 20,000 Offspring

Figure 12

Traveling Salesman Problem — 50 Cities in Three Dimensions

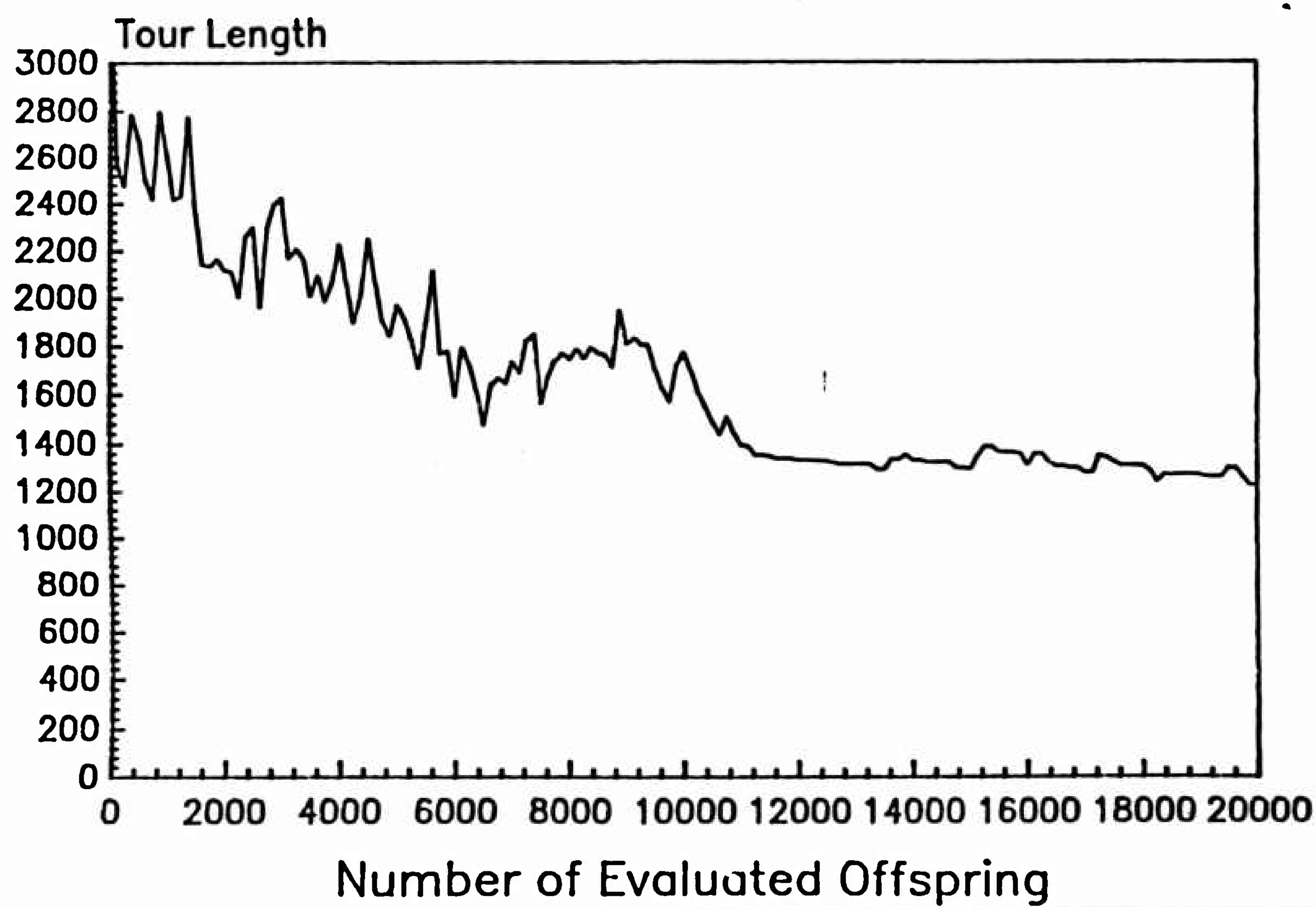


Figure 13

These experiments indicated the approximately exponential learning ability of the adaptive algorithm. However, the quality of the resulting tours remained to be determined. This can be assessed only when the distribution of the $(n-1)!/2$ tour lengths can be approximated. Two experiments were performed in this regard.

First, 28 cities were positioned in a square of perimeter 720 units. This, of course, corresponds to the optimum tour length. A computer program was written to sample 2500 tours at random and found an estimated average tour length of $\hat{\mu} = 838.8163$ (an average error of 118.8163) with an estimated standard deviation of $\sigma = 59.72235$. Because of its flexibility, a gamma function was fitted to describe the error distribution:

$$(1) \qquad f(x) = (2 \times 10^{-7})(x^3)(e^{-x/30}), \qquad x \geq 0.$$

Thirty trials were conducted with the adaptive algorithm which yielded an average tour length of 798.613 after evaluating 20,000 offspring. The average tour error was approximately 79 units. Integrating (1) from zero to 79 yields the estimated percentage of tours that were of higher quality than this average tour error. Here, this integral, calculated using Simpson's rule, was approximately 0.257. Thus, the average tour error of the adaptive algorithm was superior to roughly 75 percent of the possible tours, this after evaluating only 3.7×10^{-24} of the state space. It should be noted that 23 out of the 30 trials found perfect tours, but because the square was large, locally optimal tours had great length as compared to the optimum tour length. This dramatically increased the average length of the 30 trials.

A more complicated experiment was also performed. Here, 36 cities were organized in four groups of nine with a minimum length tour of 460 units. Again, 2500 tours were sampled at random and yielded an estimated average tour length of $\hat{\mu} = 617.1259$ (an average error of 157.1259) with a estimated standard deviation of $\sigma = 29.43465$. A gamma function was again fitted to the error distribution:

$$(2)\quad f(x) = ((5.5^{28.5})(\Gamma(28.5)))^{-1}(x^{27.5})(e^{-x/5.5}), \qquad x \geq 0.$$

Thirty trials were conducted with the adaptive algorithm which yielded an average of $\bar{x} = 518.7947$ for an average error of approximately 59 units. Integrating (2) from zero to 59 yields the estimated percentage of tours that were superior to the average results of the adaptive algorithm. This was computed to be 0.0000006438, that is to say, the adaptive algorithm produced tours that were generally superior to over 99.9999 percent of all possible tours, this after examining only 3.87×10^{-36} of the entire tour state space.

An additional set of experiments was conducted to directly compare the adaptive algorithm to the PMX operation. Here, 100 cities were distributed at random and 30 trials were performed using both methods. The cities were redistributed for each trial to minimize the effect of an unusual set of cities. Each algorithm was allowed to generate 20,000 offspring. The results were:

Adaptive Algorithm	PMX
$\bar{x} = 1454.403$ units	$\bar{x} = 4319.455$ units
$s = 110.951$ units	$s = 165.807$ units

where $\bar{x}$ is the average of the thirty trials and s is the standard deviation of the sample.

As mentioned previously, the PMX operation does not retain sufficient information between parent and offspring to perform effectively. Essentially, the PMX operation is equivalent to swapping a random number of cities in a single tour. The number of cities to be swapped is equal to the length of the section of the tour chosen at random. The expected number and variability of swaps per mutation are indicated in Figures 14 and 15. In relatively small problems, on the order of ten cities, the PMX operation averages about three swaps with minimal variance. However, in larger problems, such as the 100 city problem performed here, the PMX averages more than 33 swaps per mutation with a rather high variance. This prevents the required link between generations.

CONCLUSIONS

Successful adaptation does not require sophisticated mutations. In an evolutionary scheme only the "behavior" of a coding structure is scored; the code itself is never scored. The bottom-up view that emphasizes mutation operations as the key to adaptive plans is incorrect. Competition occurs not between coding structures but between expressed behaviors. The particular structure of the code is generally unimportant.

Further, sophisticated mutation operations can be detrimental. For adaptation to succeed, a sufficient link between parent and offspring must be maintained. When this link is destroyed the results can be worse than a random search of all possible coding structures. Since the traveling

The Expected Number of Swaps Using the PMX Operation

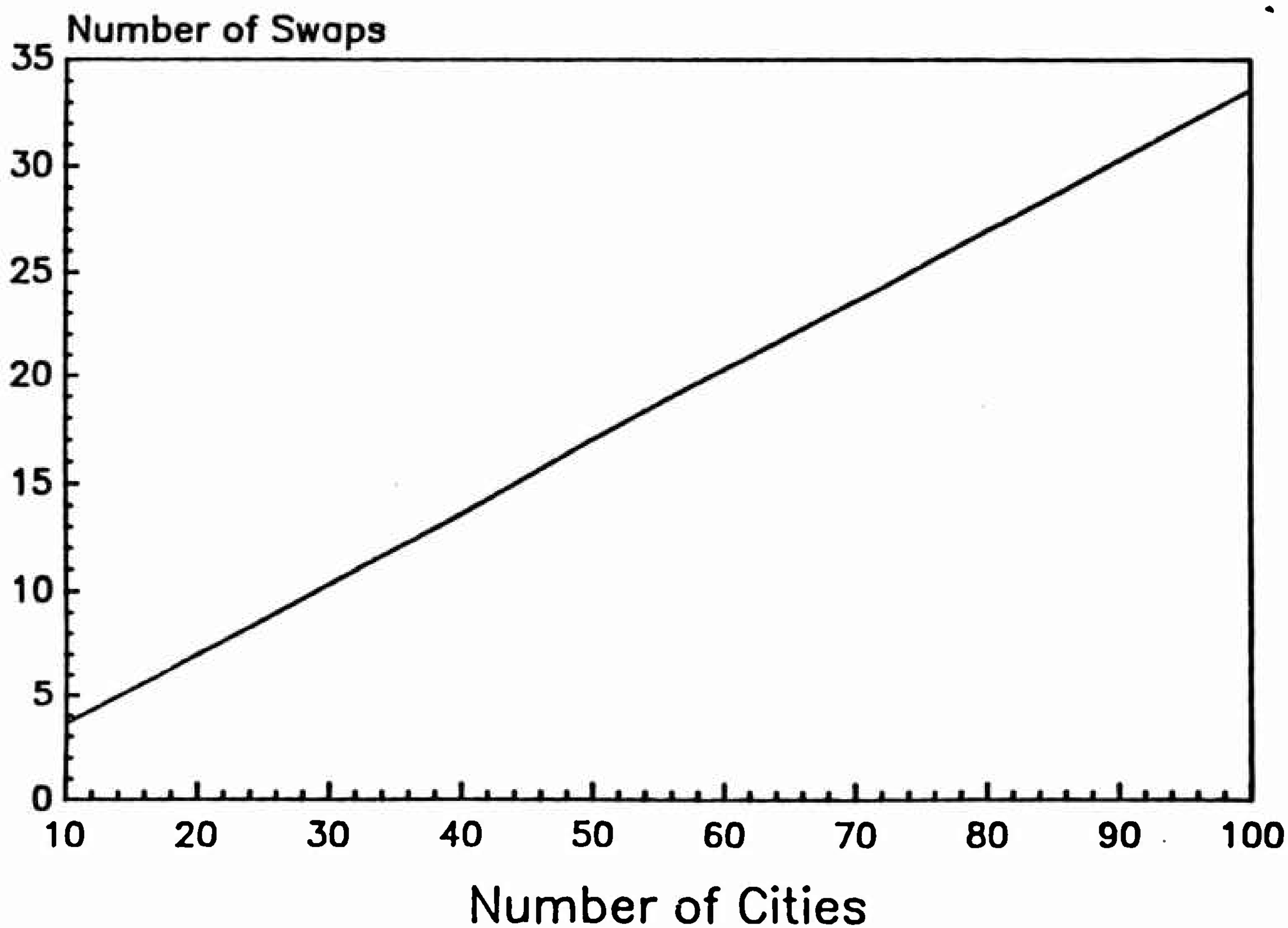


Figure 14

The Variability of the Expected Number of Swaps

Using the PMX Operation

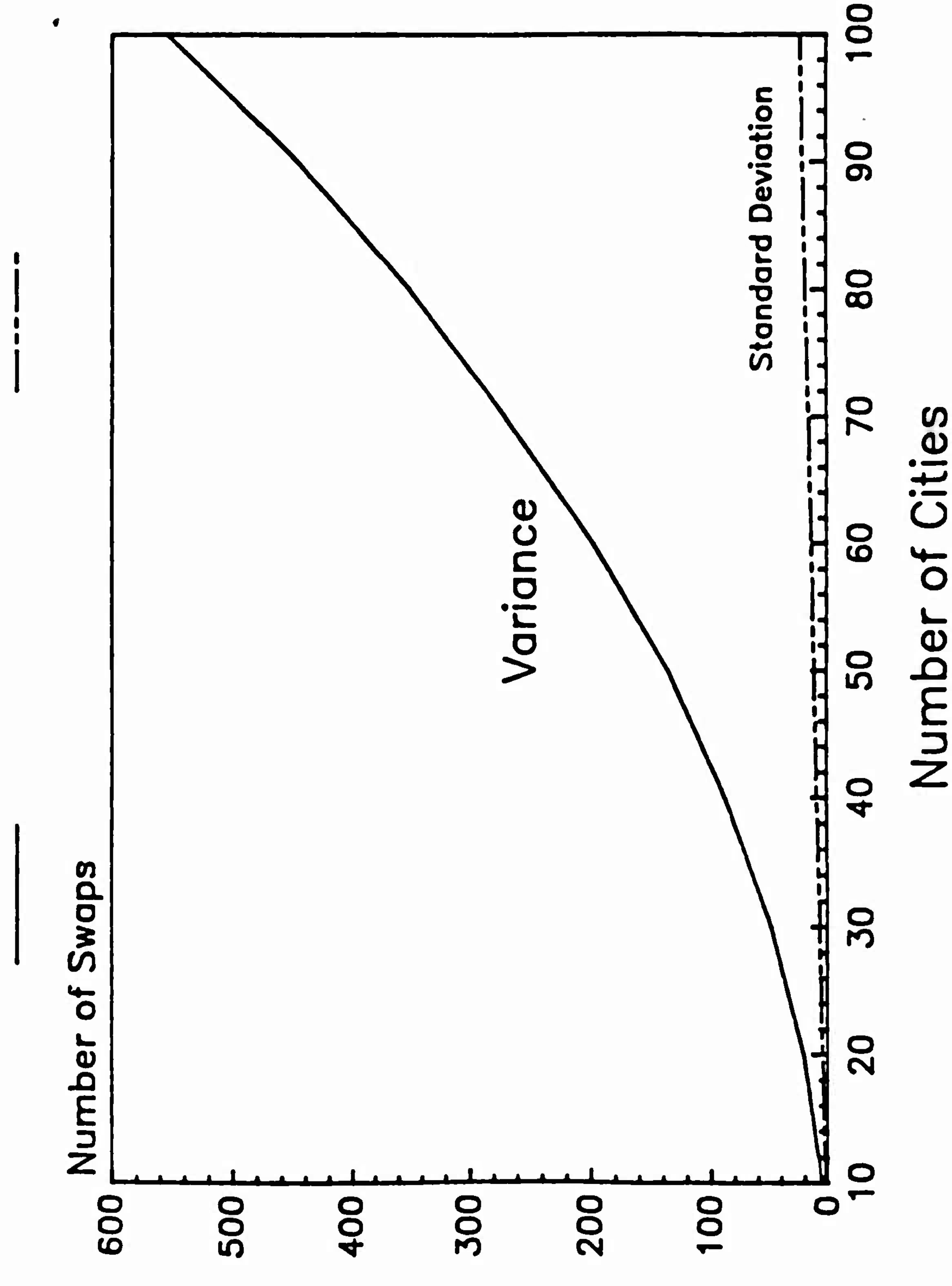


Figure 15

salesman's tour is not analogous to an organism's chromosome, operations of the form of Holland's crossover are unsuitable. In general, little emphasis should be placed on specific mutation operations. Modeling a given mutation operation in order to elicit appropriate behavior is much like requiring an airplane to possess feathers in order to fly. It is the mutation and selection of behavior that is imporatant, not the structure of the code.

Search by adaptive methods must avoid stagnation in local optima. Stagnation can be prevented through the use of a noisy payoff function. This concept is similar to that suggested by Kirkpatrick et al. (1983) for optimizing simulated annealing, but it is not necessary to resort to such specific analogies. In a dynamic environment, the rewards and penalties for different behaviors vary. The search for better and better solutions is everlasting. Evolution is a continuing process with no truly optimum solution. Incorporating noise into the adaptive algorithm prevents stagnation of the evolving phyletic line.

Clearly, evolutionary adaptation can effectively address the traveling salesman problem. The experiments described here indicate the efficiency of this evolutionary search. But, in any given problem, there is no guarantee that the optimum solution will ever be found. Evolution discovers only what it is capable of discovering. New solutions that are superior to old ones tend to survive. Despite this, the adaptive algorithm, a reification of natural evolution, tends to discover exceedingly appropriate behavior in the context of a given criteria.

Literature Cited:

1. D. Ackley, "A Connectionist Algorithm for Genetic Search," in *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, J.J. Grefenstette, Editor, Carnegie-Mellon University, July 1985.

2. A.K. Dewdney, "Computer Recreations: Exploring the field of genetic algorithms in a primordial computer sea full of flibs," *Scientific American*, November 1985.

3. L.J. Fogel, "Autonomous Automata," *Industrial Research*, February 1962.

4. L.J. Fogel, "On the Organization of Intellect," Ph.D. dissertation, U.C.L.A. 1964.

5. L.J. Fogel, A.J. Owens, M.J. Walsh, *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, New York, 1966.

6. D.E. Goldberg, R. Lingle, Jr., "Alleles, Loci, and the Traveling Salesman Problem," in *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, J.J. Grefenstette, Editor, Carnegie-Mellon University, July 1985.

7. J.J. Grefenstette, R. Gopal, B. Rosmaita, D. Van Gucht, "Genetic Algorithms for the Traveling Salesman Problem," in *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, J.J. Grefenstette, Editor, Carnegie-Mellon University, July 1985.

8. J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

9. S. Kirkpatrick, C.D. Gelatt, Jr., M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, May 1983.

10. C.K. Levy, *Elements of Biology*, third edition, Addison-Wesley Publishing Company Inc., Reading, Massachusetts, 1982.

Appendix

A completely random search (with replacement) will take roughly twice as long to find the optimum solution as an enumerative search (without replacement). To show this, consider the following two theorems:

Theorem 1: If there are β possible solutions and only one optimum solution, the expected number of trials that must be made before the optimum solution is found, using an enumerative search, assuming one trial is made at a time, is equal to $(\beta+1)/2$.

Proof: In an enumerative search, sampling is made without replacement. The probability, therefore, of discovering the optimum solution on any given trial is equal to the product of the probabilities of not discovering the optimum solution on any prior trial multiplied by the reciprical of the number of untried solutions. The expected number of trials that would have to be examined before finding the optimum solution would therefore be:

$$\begin{aligned}\Sigma\, x\cdot f(x) &= 1\cdot\beta^{-1} + 2\cdot[(\beta-1)/\beta]\cdot(\beta-1)^{-1} + 3\cdot[(\beta-1)/\beta]\cdot[(\beta-2)/(\beta-1)]\cdot(\beta-2)^{-1}\\ &\quad + \cdots + (\beta-1)\cdot[(\beta-1)/\beta]\cdots[2/3]\cdot[1/2] + \beta\cdot[(\beta-1)/\beta]\cdots[2/3]\cdot[1/2]\cdot 1\\ &= 1\cdot\beta^{-1} + 2\cdot\beta^{-1} + 3\cdot\beta^{-1} + \cdots + (\beta-1)\cdot\beta^{-1} + \beta\cdot\beta^{-1}\\ &= \beta^{-1}\cdot(1 + 2 + 3 + \cdots + (\beta-1) + \beta)\\ &= \beta^{-1}\cdot[\beta(\beta+1)/2]\\ &= (\beta+1)/2. \qquad \text{Q.E.D.}\end{aligned}$$

Theorem 2: If there are β possible solutions and only one optimum solution, the expected number of trials that must be made before the optimum solution is found, in a completely random search, assuming one trial is made at a time, is equal to β.

Proof: In a completely random search, sampling is made with replacement. The probability, therefore, of discovering the optimum solution on any given trial is equal to the product of the probabilities of not discovering the optimum solution on any previous trial multiplied by the reciprical of the total number of possible solutions. The expected number of trials that would have to be examined before finding the optimal solution would therefore be:

$$\begin{aligned}\sum x \cdot f(x) &= 1 \cdot \beta^{-1} + 2 \cdot [(\beta-1)/\beta] \cdot \beta^{-1} + 3 \cdot [(\beta-1)/\beta]^2 \cdot \beta^{-1} + \cdots \\ &= \beta^{-1} \cdot (1 + 2 \cdot [(\beta-1)/\beta] + 3 \cdot [(\beta-1)/\beta]^2 + \cdots) \\ &= \beta^{-1} \cdot [1/(1-(\beta-1)/\beta))]^2 \\ &= \beta. \qquad \text{Q.E.D.}\end{aligned}$$